Proyecto 3

Integrantes

- Acevedo Romero Miroslava
- Morales Flores Luis Enrique
- Sánchez Estrada Alejandro
- Rivera Lara Sandra Valeria

Proceso de solución de problema

Descripción del proyecto

El objetivo del proyecto es implementar el esquema de secreto compartido de Shamir.

Supongamos que tenemos un archivo super importante que queremos leer en una junta con n personas y no queremos que nadie tenga acceso a la información del documento antes de la junta en la cual se deben juntar al menos t personas para descifrar el documento. Este proyecto es un programa que cifra la información de un documento introducido en la entrada del programa y genera n claves para las n personas que participaran en la junta, el documento cifrado se sobreescribe en el documento original así que la información del documento original desaparece, cuando se requiere descifrar el programa requiere un documento con las al menos t claves y descifra el documento.

En el proceso de solución de problema se implementa el esquema de secreto compartido de shamir generando un polinomio aleatorio de grado t-1 en el cual sus coeficientes son números enteros muy grandes y el término independiente es la clave con la cual el documento fue cifrado usando el algoritmo AES, además esta clave se genera a partir de una contraseña dada usando el algoritmo SHA-256, las n claves son n evaluaciones en el polinomio, para descifrar el documento el programa recibe las claves y lee únicamente las primeras t claves para reconstruir el polinomio de Lagrange a partir de las evaluaciones del polinomio, así se recupera la clave con la cual se cifró el documento y por el último se descifra y se obtiene el documento original.

Requisitos funcionales

- Encriptado y desencriptado usando el esquema del secreto compartido de Shamir.
- El programa debe de recibir sólo los argumentos pedidos.
- El programa debe producir archivo(s) acorde a la opción seleccionada.
- Privacidad, el programa debe recibir una contraseña sin eco.

Requisitos no funcionales

- Tolerancia a fallas, el programa debe manejar los posibles errores que se puedan presentar en la entrada del programa o durante su ejecución.
- Eficiencia, el programa debe tardar la menor cantidad de tiempo posible.
- Amigabilidad, comprensible para personas que tienen nociones básicas de programación.

Entrada esperada

- Para cifrar el archivo ingrese los siguientes argumentos:
    1. Bandera c.
    2. Nombre del archivo para guardar las evaluaciones.
    3. Número total de evaluaciones requeridas (n > 2).
    4. Número mínimo de puntos necesarios para descifrar (1 < t ≤ n).
    5. Nombre del archivo a cifrar.

  Cuando el programa ejecute se solicita una contraseña sin que se pueda ver en la terminal.

- Para descifrar el archivo ingrese los siguientes argumentos:
    6. Bandera d.
    7. Nombre del archivo con las [t, n] evaluaciones.
    8. Nombre del archivo a descifrar.

Salida esperada

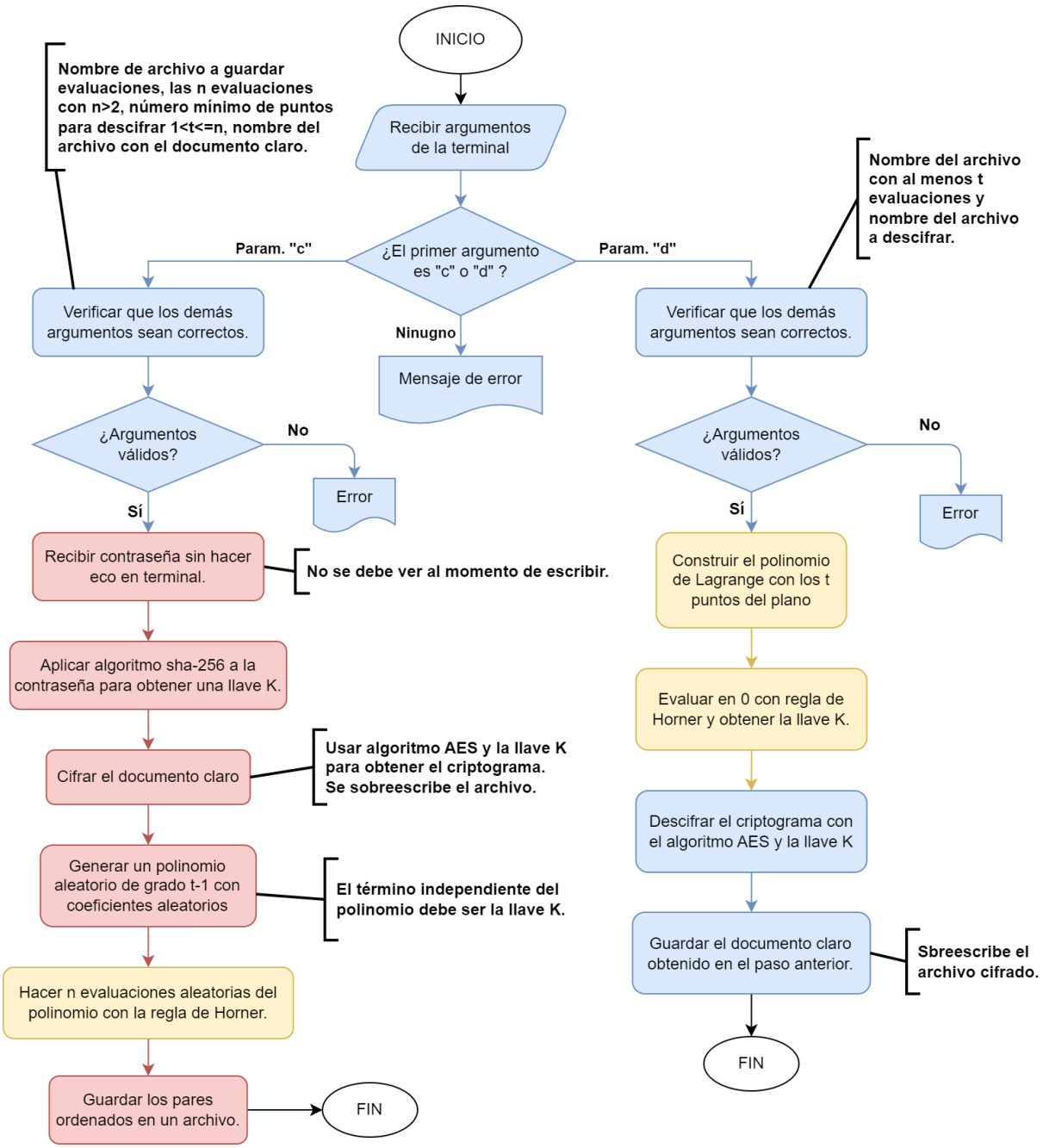
1. Al cifrar:
    - Archivo cifrado sobreescrito en el claro.
    - Archivo en donde se guardan las evaluaciones.
2. Al descifrar:
    - Archivo descifrado sobreescrito en el cifrado (texto plano original).

Herramientas

- Lenguaje: Python
- Repositorio virtual: GitHub
- Administrador de tareas: Trello
- Diagramas: diagrams.net

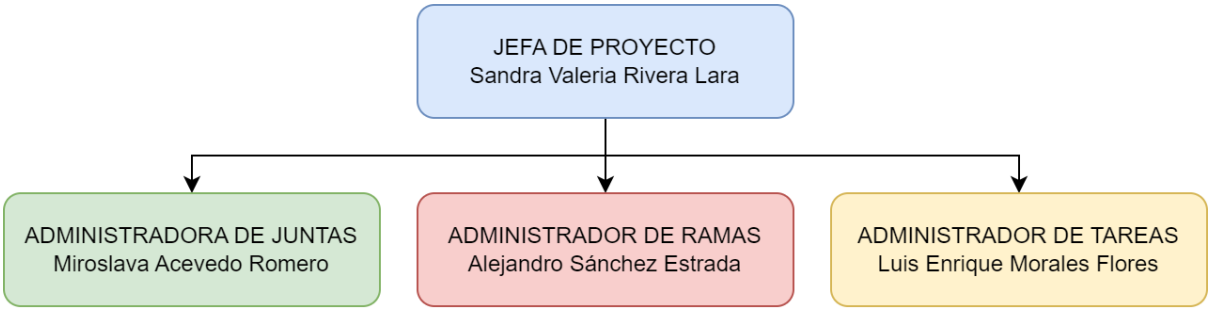
Solución

Se explican los pasos a seguir de la solución en el siguiente diagrama de flujo. Cada color representa lo que hizo cada quien, según el color del organigrama. Las pruebas unitarias no se incluyen en el diagrama de flujo.


INICIO
Recibir argumentos de la terminal
¿El primer argumento es "c" o "d" ?
Param. "c"
Param. "d"
Ninugno
Mensaje de error
Nombre de archivo a guardar evaluaciones, las n evaluaciones con n>2, número mínimo de puntos para descifrar 1<t<=n, nombre del archivo con el documento claro.
Verificar que los demás argumentos sean correctos.
¿Argumentos válidos?
No
Error
Sí
Recibir contraseña sin hacer eco en terminal.
No se debe ver al momento de escribir.
Aplicar algoritmo sha-256 a la contraseña para obtener una llave K.
Cifrar el documento claro
Usar algoritmo AES y la llave K para obtener el criptograma. Se sobreescribe el archivo.
Generar un polinomio aleatorio de grado t-1 con coeficientes aleatorios
El término independiente del polinomio debe ser la llave K.
Hacer n evaluaciones aleatorias del polinomio con la regla de Horner.
Guardar los pares ordenados en un archivo.
FIN
Nombre del archivo con al menos t evaluaciones y nombre del archivo a descifrar.
Verificar que los demás argumentos sean correctos.
¿Argumentos válidos?
No
Error
Sí
Construir el polinomio de Lagrange con los t puntos del plano
Evaluar en 0 con regla de Horner y obtener la llave K.
Descifrar el criptograma con el algoritmo AES y la llave K
Guardar el documento claro obtenido en el paso anterior.
Sbreescribe el archivo cifrado.
FIN

Organización del equipo

Organigrama



Participación de los integrantes

Jefa de proyecto (Sandra):

- Aporte: Leer la entrada, revisar los argumentos, descifrar y crear el archivo. Además de crear respuestas a excepciones durante la ejecución.
- Rama: entrada, debug (apoyo).

Administradora de juntas (Miroslava)

- Aporte: Pruebas unitarias, resumen de juntas.
- Rama: pruebas-unitarias.

Administrador de ramas (Alejandro):

- Aporte: Cifrado AES, llave k con sha-256, Evaluación del polinomio.
- Rama: Cifrado_Shamir.

Administrador de tareas (Luis Enrique):

- Aporte: Regla de Horner, reconstrucción del polinomio de Lagrange y sus respectivas pruebas unitarias.
- Rama: descifrarPolinomio y debug.

Juntas

Junta 1

Temas a tratar junta 1

- Herramientas de trabajo (lenguaje) (Python)
- Diagrama de flujo (solución de problemas)
- Asignación de tareas

- Creación de repositorio y Trello

Resumen junta 1

Para la primera junta para el proyecto 3 se había acordado mantener los roles de cada miembro del equipo como en el proyecto 2. De acuerdo con los temas decididos para tratarse en la junta, se inició confirmando el lenguaje de programación del proyecto y se analizó el documento donde se explicaba el programa. El administrador de tareas presentó un esbozo del diagrama de flujo para clarificar la división del problema y las tareas derivadas para los integrantes. También se identificaron posibles problemas con ciertas partes del algoritmo, algunos de los cuales se presentan en los pendientes para la siguiente junta de proyecto. Se calendarizó la siguiente junta de proyecto para el 26 de noviembre y se dividieron las tareas como se presenta a continuación:

Repartición de tareas iniciales

- Sandra: entrada, salida
- Miros: pruebas unitarias
- Luke: polinomio (descifrado), regla de Horner
- Alejo: cifrado (desde algoritmo sha, cifrado con AES y llave, creación del polinomio) (azul fuerte y rojo)

Pendientes para siguiente junta (26/11)

- Preguntar a Ximena o Galaviz acerca de los números enteros y:
  - Se requiere que los resultados sean números enteros. Así que habrá que trabajar en un campo finito.
  - Se requiere entonces de operar siempre módulo algún primo grande, de 256 bits o cerca.
- Creación del repositorio con restricciones de seguridad (Alejo)
- Trello (Luke)
- Horner (Luke)
- Versión 1 del pdf

Junta 2

Temas a tratar junta 2

- Avances del proyecto
- Números enteros
- Código de polinomios

Resumen junta 2

La junta 2 se pospuso para el lunes 27 de noviembre a las 19:00. Al iniciar, la jefa del proyecto expuso su parte (entrada) en funcionamiento y se aclararon dudas acerca de ésta y de su integración con la sección de descifrado. Luego, el administrador de ramas expuso su parte (cifrado) y quedó pendiente rectificar su correcto funcionamiento, así como integrarlo con la entrada. Se estableció una fecha para la siguiente junta del proyecto (3 de diciembre) y los avances requeridos para esa fecha (descifrado e integración de las partes existentes).

Pendientes para siguiente junta (3/12)

- Integrar partes de Alejo y Sandra
- Avances de parte de Luke

Junta 3

Temas a tratar junta 3

- Avances (Luke)
- Estado de las ramas
- Integración de las partes

Resumen junta 3

La junta 3 se pospuso el mismo día a las 8:30 p.m. Lo primero que se trató fueron los avances de la parte de descifrado de polinomios y el método de evaluación usado para estos tanto en la parte del cifrado como en la de descifrado (método de Horner). Tras haber discutido el funcionamiento se llegó a la conclusión de que los módulos probablemente están listos para acoplarse entre sí, y esta tarea quedó pendiente. Se añadió el código del descifrado al repositorio durante la junta, y con esto también quedó pendiente hacer las pruebas para los módulos.

Pendientes

- Integrar partes de Luke y Alejo (Sandra/Alejandro) para probar programa en main.
- Completar pruebas unitarias de los módulos separados (empezando por manejo de errores de entrada y salida) (Sandra)